**Report on Quine McCluskey project**

**Name: Youssef Mohamed Alshaarawi**

**ID: 80**

**Language used: C++**

**Problem statement:**

*Make a program to minimize any Boolean function in S.O.P*

*Form using Quine McCluskey’s method.*

**Input:**

* *Number of variables*
* *Number of Minterms*
* *Minterms*
* *Number of don’t care*
* *Don’t care*

**Output:**

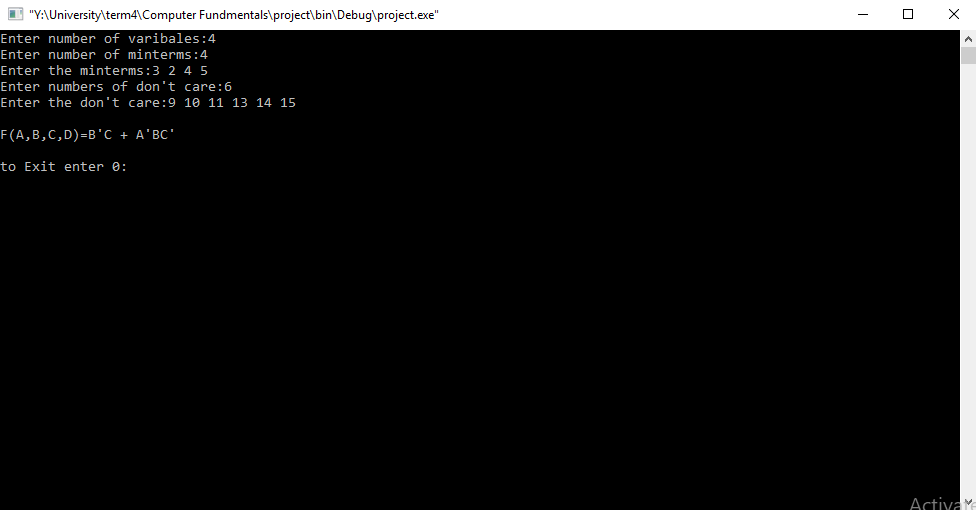
* *Minimized Function Expression*

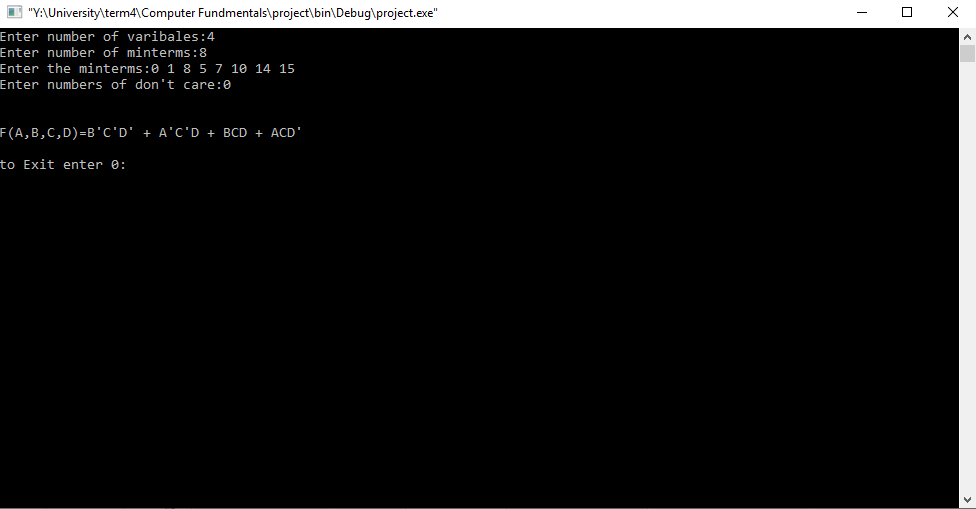
**Methodology Used:**

* *Tabular Method*
* *Prime Implicates table*
* *Remove essentials prime implicates*
* *Check* ***if*** *all minterms covered*
* *Print the function*
* *else Row Dominance*
* *Remove essentials prime implicates*
* *Check* ***if*** *all minterms covered*
* *Print the function*
* *Column dominance*
* *Remove essentials prime implicates*
* *Check* ***if*** *all minterms covered*
* *Print the function*
* *else Row Dominance*
* *Remove essentials prime implicates*
* *Check* ***IF*** *all minterms covered*
* *Print the function*
* *Else Patrick’s method*
* *Print the function*

**Used Data Structures:**

* Vectors:
  1. *Store Prime Implicates*
  2. *Store Prime Implicates table*
  3. *Store essentials*
  4. *Store function after minimization*
* Pairs:
  1. *Store every minterm and whether it is covered or not*
* Map:
  1. *Store number of ones for every minterm in their binary from*
* Set:
  1. *Used to remove repetition of Prime implicates in Patrick’s method*

**Sample Runs:****

**